Towards generic and efficient constraint-based mining, a constraint programming approach

Tias Guns

KU Leuven, Celestijnenlaan 200A, Leuven, Belgium

tias.guns@cs.kuleuven.be

http://people.cs.kuleuven.be/~tias.guns/

Abstract

In today's data-rich world, pattern mining techniques allow us to extract knowledge from data. However, such knowledge can take many forms and often depends on the application at hand. This calls for generic techniques that can be used in a wide range of settings. In recent years, constraint programming has been shown to offer a generic methodology that fits many pattern mining settings, including novel ones. Existing constraint programming solvers do not scale very well though. In this talk, I will review different ways in which this limitation has been overcome. Often, this is through principled integration of techniques and data structures from pattern mining into the constraint solvers.

The fields of data mining and constraint programming are amongst the most successful subfields of artificial intelligence. Yet, their methodologies are quite different. Constraint programming advocates a declarative modeling and solving approach to constraint satisfaction and optimisation problems. Data mining on the other hand has focussed on handling large and complex datasets that arise in particular applications. Pattern mining more specifically aims to extract interesting patterns from a dataset, where *interestingness* is often defined by the application at hand. Current ad-hoc methods often focus on special-purpose algorithms to specific problems and interestingness criteria. This typically yields complex code that is very efficient, but hard to modify or reuse in other applications. Hence, less attention has been devoted to the issue of general and generic solution strategies.

Nevertheless, there is a need for generic techniques that can handle variations of known tasks, as well as application-driven constraints Dzeroski et al. (2010); De Raedt et al. (2011). The typical iterative nature of the knowledge-discovery cycle Han and Kamber (2000), in which the data and problem definition are iteratively defined based on prototyping and small scale evaluations. In this case, the problem specification typically changes between iterations, which may in turn require changes to the algorithms.

This is acknowledged in the field of *constraint-based mining*, which adopts the methodology of formulating a problem in terms of constraints Nijssen (2010); Boulicaut and Jeudy (2005). For example, for itemset mining Agrawal et al.

(1993), a wide variety of other constraints and a range of algorithms for solving these constraint-based itemset mining problems Mannila and Toivonen (1997); Jr. et al. (2000); Pei and Han (2000); Pei et al. (2001); Bucila et al. (2003); Han et al. (2007); Soulet and Crmilleux (2005); Bonchi and Lucchese (2007) has enabled the application of itemset mining to numerous other problems, ranging from web mining to bioinformatics Han et al. (2007). Generic frameworks in the constraint-based mining literature have focussed on the (anti-)*monotonicity* of constraints Mannila and Toivonen (1997); Pei and Han (2000); Bucila et al. (2003) leading to systems such as ConQueSt Bonchi and Lucchese (2007), MusicDFS Soulet and Crmilleux (2005) and Molfea De Raedt and Kramer (2001). While many typical data mining tasks consist of (anti-)monotonic constraints, many other constraints do not fit in this framework, such as finding *closed* patterns in dense data Pasquier et al. (1999); Pei et al. (2000), or mining for correlated patterns in supervised data Morishita and Sese (2000); Cheng et al. (2007). Frameworks that are more generic than (anti-) monotonicity and in which arbitrary combinations of constraints are allowed have been missing.

Constraint programming and itemset mining The CP4IM framework De Raedt et al. (2008); Guns et al. (2011a) was the first to propose a generic CP-based framework for constraint-based itemset mining. The framework encompassed frequent itemset and constraints ranging from typical (anti-) monotone constraints such as size and cost of the pattern, as well as *condensed* representation constraints such as closed and maximal.

Since then, many different works have extended this approach, including:

- Use of different declarative solving techniques. Other techniques explored include knowledge compilation and BDDs (Cambazard et al., 2010), Answer Set Programming (Järvisalo, 2011) and SAT solving (Jabbour et al., 2015; Coquery et al., 2012);
- Pattern *set* mining, also known as n-ary patterns. Here the goal is not to find all individual patterns, but rather to find a concise set of n patterns (Guns et al., 2011b; Khiari et al., 2010);
- Optimisation and top-k mining. Also here the goal is not to enumerate all satisfying patterns but rather to find the optimal pattern, e.g. according to a measure of correlation or discrimination Nijssen et al. (2009), or to find the top-k most optimal patterns Jabbour et al. (2013).
- Multi-objective optimisation, also known as mining skypatterns. In this case multiple measures are given and the Pareto-optimal solutions are sought Kemmar et al. (2014); Rojas et al. (2014). A generalisation from multi-objective to *dominance* relations also encompasses condensed representations such as closed/maximal pattern mining and finding relevant subgroups Negrevergne et al. (2013).

Interestingly, most of these approaches use unmodified solvers and are still able to obtain reasonable efficiency, especially in the case when many constraints are present Guns et al. (2011a). The key low-level constraint that these formulations use is a *reified* weighted sum constraint over Boolean variables, where reified means that the truth value of the constraint is reflected in a Boolean indicator variable. Notable in this respect is that one can implement a CP solver

over Boolean variables using the data structures used in itemset mining algorithms, and achieve the same scalability as depth-first itemset mining, while also having the same generality as other CP solvers have for itemset mining Nijssen and Guns (2010).

Constraint programming and sequence mining A sequential pattern is an ordered list of events. This sequential ordering differs from the traditional (unordered) interpretation of an itemset pattern. Furthermore, the same event can reoccur multiple times in a sequential pattern.

A key property of any pattern mining method is the ability to compute the *frequency* of a pattern; this consists of verifying for each entry in the database whether the pattern occurs in this entry (such an entry is often called a *transaction*. For itemsets, this corresponds to verifying that the pattern is a subset of the transaction, and for sequences that it is a subsequence of the transaction.

Works that use constraint programming for constraint-based mining can be divided into two camps, based on the representation of a sequence:

- Sequences with explicit wildcards. An example is $\langle A, *, B\rangle$ where $*$ is the wildcard. This will match any transaction that contains an A, followed by a single arbitrary event, followed immediately by event B. It would not match with a transaction such as $\langle A, C, C, B\rangle$, but it would match with $\langle C, A, C, B\rangle$. This problem can be formulated in a way that is very similar to frequent itemset mining Coquery et al. (2012) and hence many of the same constraints and variations can be expressed, including frequent, closed and maximal Coquery et al. (2012) and top-k and relevant subgroups Kemmar et al. (2014).

- Sequences with implicit wildcards. This is the more traditional sequence pattern considered, where a pattern $\langle A, B\rangle$ is a subsequence of all of $\langle C, A, B\rangle$, $\langle A, C, C, B\rangle$ and $\langle A, C, \ldots, C, B\rangle$ as there are implicit wildcards between all symbols. This is much more difficult to express in a constraint solver Métivier et al. (2013) as in the general case, testing the subsequence relation for an individual transaction requires *searching* over all possible matchings, which is worst-case exponential. Two ways to overcome this are first, to add a global constraint that does this transparently to the CP solver, and second to decompose the subsequence constraint and treat it for each transaction as an independent subproblem that requires search Negrevergne and Guns (2015). The former approach is most efficient as the same prefix-projection technique as used in PrefixSpan Han et al. (2001) can be used, including pruning of infrequent extensions. Even better scalability can be obtained by having one global constraint that does this for all transactions at once, instead of having one separate constraint for each transaction Kemmar et al. (2015).

The work on sequences shows us that itemsets are quite exceptional in that all constraints, including condensed representations, can be expressed using standard constraints available in CP. Only top-k, multi-objective optimisation and *dominance* relations require changes to the solving procedure. On the other hand, to model sequence while obtaining reasonable solving performance specialised constraints or search procedures need to be written. Furthermore, hiding the *subsequence* check within a global constraint is most efficient but

does not allow to change the subsequence relation, e.g. to enforce a maximum gap between matching elements, without changing the code implementing the constraint. There is hence still room for truly generic techniques for sequence mining, as well as for other structured pattern mining tasks such as graph mining. See Guns et al. (2016) for a more detailed discussion of the challenges and possible solutions.

A language for generic pattern mining? Developing generic languages for pattern mining is a long standing quest Bonchi and Lucchese (2007); Soulet and Crmilleux (2005); Blockeel et al. (2012); Métivier et al. (2012); Guns et al. (2013a). Many efforts have their roots in the idea of inductive databases Mannila (1997); these are databases in which both data and patterns are first-class citizens and can be queried. Most inductive query languages, e.g., Meo et al. (1996); Imielinski and Virmani (1999), extend SQL with primitives for pattern mining. They have only a restricted language for expressing mining problems, and are usually tied to one mining algorithm. A more advanced development is that of mining views Blockeel et al. (2012), which provides lazy access to patterns through a virtual table. Standard SQL can be used for querying, and the implementation will only materialize those patterns in the table that are relevant for the query. This is realized using a traditional mining algorithm.

More recent work has looked at high-level languages that have a straightforward translation into a declarative specification of the problem Guns et al. (2013b); Métivier et al. (2012). At the same, many high-level modeling languages exist in the constraint programming literature Van Hentenryck (1999); Van Hentenryck and Michel (2005); Frisch et al. (2008); Nethercote et al. (2007).

The MiningZinc system Guns et al. (2013a) unifies both approaches by adopting the MiniZinc constraint programming language Nethercote et al. (2007), while at the same time offering additional abstractions that often occur in pattern mining problems. The language is independent of any solving technology which gives the MiningZinc system the ability to verify whether an existing algorithm exists that matches the problem formulation, or whether a generic constraint solver should be used. In the former case, a highly efficient and scalable specialized algorithm can be used. Furthermore, using rewrite rules, the system can detect that a specialized can be used to solve part of the problem, and that the remaining constraints can be post-processed. The result is a hybredisation of solving techniques, all of which is hidden behind a high-level generic language.

Conclusions In this talk and accompanying paper I have highlighted recent advances on bridging the methodological gap between the fields of data mining and constraint programming. The over aching goal is make data mining approaches more flexible and declarative, so as to make it easy to change the model without requiring reimplementation work on the solver. Indeed, many of the referenced approaches are more generic than existing systems.

On the other hand, there is often a tradeoff between generality and efficiency, and devising methods that are both generic and scalable is the prime challenge. Many recent successes have in some way hybridized data structures or algorithms from specialized methods into generic constraint solvers. This is a very promising approach that brings the data mining and constraint programming

fields closer not only at the application level but also at the algorithmic level.

Acknowledgments

I would like to thank Luc De Raedt and Siegfried Nijssen for the many fruitful collaborations and discussions, and colleagues Anton Dries, Benjamin Negrevergne and Behrouz Babaki who helped shape and develop these ideas. This work is also the result of many interesting papers and discussions with other researchers at workshops and conferences.

References

Agrawal, R., Imielinski, T., and Swami, A. N. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press.

Blockeel, H., Calders, T., Fromont, É., Goethals, B., Prado, A., and Robardet, C. (2012). An inductive database system based on virtual mining views. *Data Min. Knowl. Discov.*, 24(1):247–287.

Bonchi, F. and Lucchese, C. (2007). Extending the state-of-the-art of constraint-based pattern discovery. *Data Knowl. Eng.*, 60(2):377–399.

Boulicaut, J.-F. and Jeudy, B. (2005). Constraint-based data mining. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, pages 399–416. Springer US.

Bucila, C., Gehrke, J., Kifer, D., and White, W. M. (2003). Dualminer: A dual-pruning algorithm for itemsets with constraints. *Data Min. Knowl. Discov.*, 7(3):241–272.

Cambazard, H., Hadzic, T., and O'Sullivan, B. (2010). Knowledge compilation for itemset mining. In *19th European Conference on Artificial Intelligence*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 1109–1110. IOS Press.

Cheng, H., Yan, X., Han, J., and Hsu, C.-W. (2007). Discriminative frequent pattern analysis for effective classification. In *Proceedings of the 23rd International Conference on Data Engineering*, pages 716–725. IEEE.

Coquery, E., Jabbour, S., Saïs, L., and Salhi, Y. (2012). A sat-based approach for discovering frequent, closed and maximal patterns in a sequence. In *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, pages 258–263.

De Raedt, L., Guns, T., and Nijssen, S. (2008). Constraint programming for itemset mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-08)*, pages 204–212. ACM.

De Raedt, L. and Kramer, S. (2001). The levelwise version space algorithm and its application to molecular fragment finding. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 853–862. Morgan Kaufmann.

De Raedt, L., Nijssen, S., O'Sullivan, B., and Van Hentenryck, P. (2011). Constraint programming meets machine learning and data mining (dagstuhl seminar 11201). *Dagstuhl Reports*, 1(5):61–83.

Dzeroski, S., Goethals, B., and Panov, P. (2010). *Inductive Databases and Constraint-Based Data Mining*. Springer-Verlag New York, Inc., 1st edition.

Frisch, A., Harvey, W., Jefferson, C., Hernández, B. M., and Miguel, I. (2008). Essence: A constraint language for specifying combinatorial problems. *Constraints*, 13(3):268–306.

Guns, T., Dries, A., Tack, G., Nijssen, S., and De Raedt, L. (2013a). MiningZinc: A modeling language for constraint-based mining. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, International Joint Conference on Artificial Intelligence, Beijing, China, 3-9 August 2013*, pages 1365–1372. AAAI Press.

Guns, T., Nijssen, S., and De Raedt, L. (2011a). Itemset mining: A constraint programming perspective. *Artif. Intell.*, 175(12-13):1951–1983.

Guns, T., Nijssen, S., and De Raedt, L. (2011b). k-pattern set mining under constraints. *IEEE Transactions on Knowledge and Data Engineering*, To Appear. Also as Technical Report CW596, Oct 2010.

Guns, T., Nijssen, S., and De Raedt, L. (2013b). k-Pattern set mining under constraints. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):402–418.

Guns, T., Paramonov, S., and Negrevergne, B. (2016). On declarative modeling of structured pattern mining. In *AAAI Workshop on Declarative Learning Based Programming, Phoenix, Arizona USA, 12-13 February 2016*. Accepted.

Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data Min. Knowl. Discov.*, 15(1):55–86.

Han, J. and Kamber, M. (2000). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. (2001). Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth. *ICDE'2001, April*, pages 215–24.

Imielinski, T. and Virmani, A. (1999). MSQL: A query language for database mining. *Data Mining and Knowledge Discovery*, 3:373–408.

Jabbour, S., Sais, L., and Salhi, Y. (2013). The top-k frequent closed itemset mining using top-k SAT problem. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, pages 403–418.

Jabbour, S., Sais, L., and Salhi, Y. (2015). Decomposition based SAT encodings for itemset mining problems. In *Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part II*, pages 662–674.

Järvisalo, M. (2011). Itemset mining as a challenge application for answer set enumeration. In *Logic Programming and Nonmonotonic Reasoning - 11th International Conference, LPNMR 2011, Vancouver, Canada, May 16-19, 2011. Proceedings*, pages 304–310.

Jr., R. J. B., Agrawal, R., and Gunopulos, D. (2000). Constraint-based rule mining in large, dense databases. *Data Min. Knowl. Discov.*, 4(2/3):217–240.

Kemmar, A., Loudni, S., Lebbah, Y., Boizumault, P., and Charnois, T. (2015). PREFIX-PROJECTION global constraint for sequential pattern mining. In *Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings*, pages 226–243.

Kemmar, A., Ugarte, W., Loudni, S., Charnois, T., Lebbah, Y., Boizumault, P., and Crémilleux, B. (2014). Mining relevant sequence patterns with cp-based framework. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014, Limassol, Cyprus, November 10-12, 2014*, pages 552–559.

Khiari, M., Boizumault, P., and Crémilleux, B. (2010). Constraint programming for mining n-ary patterns. In *Principles and Practice of Constraint Programming*, volume 6308 of *Lecture Notes in Computer Science*, pages 552–567. Springer.

Mannila, H. (1997). Inductive databases and condensed representations for data mining. In *ILPS*, pages 21–30.

Mannila, H. and Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Min. Knowl. Discov.*, 1(3):241–258.

Meo, R., Psaila, G., and Ceri, S. (1996). A new SQL-like operator for mining association rules. In *VLDB*, pages 122–133.

Métivier, J., Boizumault, P., Crémilleux, B., Khiari, M., and Loudni, S. (2012). A constraint language for declarative pattern discovery. In *Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Riva, Trento, Italy, March 26-30, 2012*, pages 119–125.

Métivier, J.-P., Loudni, S., and Charnois, T. (2013). A constraint programming approach for mining sequential patterns in a sequence database. In *ECML/PKDD 2013 Workshop on Languages for Data Mining and Machine Learning*. also available as arXiv:1311.6907.

Morishita, S. and Sese, J. (2000). Traversing itemset lattice with statistical metric pruning. In *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 226–236. ACM.

Negrevergne, B., Dries, A., Guns, T., and Nijssen, S. (2013). Dominance programming for itemset mining. In *13th IEEE International Conference on Data Mining, IEEE International Conference on Data Mining, Dallas, Texas, USA, 7-10 December 2013*, pages 557–566. IEEE Computer Society.

Negrevergne, B. and Guns, T. (2015). Constraint-based sequence mining using constraint programming. In *Integration of AI and OR Techniques in Constraint Programming, Barcelona, Spain, 18-22 May 2015*. Springer International Publishing. Accepted.

Nethercote, N., Stuckey, P. J., Becket, R., Brand, S., Duck, G. J., and Tack, G. (2007). MiniZinc: Towards a standard CP modelling language. In *CP*, volume 4741 of *LNCS*, pages 529–543. Springer.

Nijssen, S. (2010). Constraint-based mining. In Sammut, C. and Webb, G., editors, *Encyclopedia of Machine Learning*, pages 221–225. Springer US.

Nijssen, S. and Guns, T. (2010). Integrating constraint programming and itemset mining. In *Machine Learning and Knowledge Discovery in Databases, European Conference (ECML/PKDD-10)*, volume 6322 of *Lecture Notes in Computer Science*, pages 467–482. Springer.

Nijssen, S., Guns, T., and De Raedt, L. (2009). Correlated itemset mining in ROC space: A constraint programming approach. In *KDD*, pages 647–656. ACM.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *Database Theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 398–416. Springer.

Pei, J. and Han, J. (2000). Can we push more constraints into frequent pattern mining? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 350–354. ACM.

Pei, J., Han, J., and Lakshmanan, L. V. S. (2001). Mining frequent item sets with convertible constraints. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 433–442. IEEE.

Pei, J., Han, J., and Mao, R. (2000). Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30. ACM.

Rojas, W. U., Boizumault, P., Loudni, S., Crémilleux, B., and Lepailleur, A. (2014). Mining (soft-) skypatterns using dynamic CSP. In *Integration of AI and OR Techniques in Constraint Programming - 11th International Conference, CPAIOR 2014, Cork, Ireland, May 19-23, 2014. Proceedings*, pages 71–87.

Soulet, A. and Crmilleux, B. (2005). An efficient framework for mining flexible constraints. In *Advances in Knowledge Discovery and Data Mining*, volume 3518 of *Lecture Notes in Computer Science*, pages 43–64. Springer.

Van Hentenryck, P. (1999). *The OPL optimization programming language*. MIT Press.

Van Hentenryck, P. and Michel, L. (2005). *Constraint-Based Local Search*. MIT Press.